REMARKS

In the Office Action mailed June 14, 2007, claims 1-14 and 21-27 were rejected. Claims 1-14 and 21-27 were rejected under 35 U.S.C. §112, ¶2 as being indefinite. Claims 1-14 and 21-27 were also rejected under 35 U.S.C. §103(a) as being obvious over Bengston (U.S. Pat. No. 6,728,947) in view of Nichols et al. (U.S. Pat. No. 6,018,730) in further view of Bacon et al. (U.S. Pat. No. 6,430,538).

Interview Summary

A telephonic interview was conducted on August 20, 2007 between inventor Kuldeep Dhar and Austen Zuege (Reg. No. 57,907) for Applicants and Examiner Kalyan Deshpande. During the interview, a demonstration of working model of the invention was given via the Internet by inventor Kuldeep Dhar. In addition, the arguments presented in the Amendment filed on March 19, 2007 and the §112 rejections of claims 1-14 and 21-27 as stated on pages 2-4 of Office Action were discussed. Independent claims 1 and 8 were briefly discussed in conjunction with the §103(a) rejections over Bengston in view of Nichols et al. in further view of Bacon et al. Examiner Deshpande indicated that rewording of the "runtime setup" language introduced in the Amendment filed on March 19, 2007 could overcome the §112 rejections. Furthermore, Examiner Deshpande noted that he had not seen numerous features of the invention as demonstrated with the working model before in the prior art, such as the ability to use data dictionaries configurable for different industry applications without reprogramming effort, task processing configuration (such as timing) using business parameter objects (called "task association icons" in the working model), or administrative tools for process management and optimization (referred to in the demonstration as a "simulator" and "validation" functions). However, the Examiner recommended that the claim language be amended to further clarify the unique functionality of the present invention. No agreement was reached regarding the claims, with regard to either the §112 or §103(a) rejections.

Claim Rejections - 35 U.S.C. §112, ¶2

Claims 1-14 and 21-27 were rejected under 35 U.S.C. §112, ¶2 as being indefinite with regard to the claim term "runtime setup". With the present amendment, that language has been amended, such that software programming occurring before runtime (e.g., programming effort by software developers) is more clearly distinguished from workflow configuration performed by business users (e.g., lenders) during runtime, as well as from the execution of configured workflow task lists or checklists during runtime (e.g., the processing of a loan application). It should further be noted that configuration effort by business users at runtime according to the present invention, for example to configure a particular workflow checklist or task list, does not disrupt the software components defined prior to runtime. Rather, the software components defined prior to runtime are made available for business users to utilize in configuring workflow checklists or task lists during runtime, for instance, where business users drag and drop symbols that correspond to the defined software components using a workspace in a graphical interface.

With regard to arguments made in the Amendment filed on March 19, 2007, Applicants note that the intent of those arguments was to indicate that Bengston and Nichols et al. are fixed systems with all software components as well as the overall workflow processing organization (i.e., the checklist or task list organization) is defined prior to runtime. In contrast, Bacon et al. appears to lack software components like tasks that are defined prior to runtime, but instead all definitions of workflow tasks are delayed until runtime. The present invention does not fall entirely within either of those categories, but rather is a novel and non-obvious hybrid, utilizing software components (e.g., tasks) defined prior to runtime and checklists or task lists that are configured during runtime, which allows numerous advantages over the prior art.

Thus, it is believed that with the present amendments, all of claims 1-14 and 21-27 are definite. The rejections of those claims under §112, ¶2 should be withdrawn, and notification to that effect is requested.

Claim Rejections - 35 U.S.C. §103(a)

Claims 1-14 and 21-27 were rejected under 35 U.S.C. §103(a) as being obvious over Bengston (U.S. Pat. No. 6,728,947) in view of Nichols et al. (U.S. Pat. No. 6,018,730) in further view of Bacon et al. (U.S. Pat. No. 6,430,538). A detailed overview of the present invention was provided in the March 19, 2007 Amendment, and was discussed in the interview conducted August 20, 2007. Therefore, an overview of the invention is omitted here for simplicity. However, the relevant claims as presently amended are discussed in turn.

Amended independent claim 1 recites a workflow management system for hosting process-based tasks and decisioning with a collection of software components on a single platform that includes a software component for business users to establish configurable workflow checklists in real-time in which a plurality of differentiated tasks are set up and are made available for configuring any type of workflow and each workflow task can avail of a plurality of existing or new underlying business parameter objects that can be embedded for workflow task automation, and a data dictionary associated with each workflow. Each workflow is driven by the associated data dictionary for a selected industry to which that workflow corresponds, and the software component for business users has the ability to use, handle and manage the data dictionary and to generate entry conditions and rules dynamically without restarting applications or rewriting underlying software code. The software component for business users includes a graphical interface usable to configure workflows at runtime, which follows a software programming stage, the graphical user interface having a list of business parameter objects represented as geometric shapes and a workspace, each business parameter object represented as a geometric shape being an abstracted object-based representation of functions within the collection of software components, the workspace for organizing and linking multiple geometric shapes at runtime in an ordered arrangement of objects, the ordered arrangement of objects corresponding to an order in which the multiple differentiated tasks are performed when any of the configurable workflow checklists are executed. The system according to amended independent claim 1 further includes a database for storing the arrangement of objects in the configurable workflow checklists as well as for storing the entry conditions and

embedding information for the business parameter objects that are associated with each of the multiple differentiated tasks.

Amended independent claim 8 recites a workflow system for programmatically managing dynamic workflow processes that includes a workflow engine, a workflow designer, a rules database, and a data dictionary. The workflow engine is for performing task list processing as defined by a plurality of task lists, with any number of the plurality of task lists processed by the workflow engine at any given time, and is a software component containing a plurality of discrete functions defined for each application within the workflow system prior to runtime. The a workflow designer is for configuring a plurality of task lists and has an object-based interface for drag-and-drop creation and modification of task lists at runtime. The workflow designer having a display window that includes a function list containing multiple symbols each corresponding to at least one of the plurality of discrete functions accessible within the workflow engine at runtime, a business parameter object list with each business parameter object therein able to be embedded with any of the discrete functions represented as symbols, a workspace providing a graphical area for assembly of ordered task lists at runtime, and tools for configuring entry conditions associated with any of the plurality of discrete functions for each task list according to logical mathematical operators selected from the rules database and configured at runtime. The workflow designer allows for assembly of ordered tasks by dragging and dropping one of the multiple symbols into the workspace, and embedding business parameter objects with any of the discrete functions represented as symbols, with the workflow designer providing graphical links for assembling and reassembling an ordered task list from multiple discrete symbols. The data dictionary is configurable for each task list for defining discrete data elements and data relationships that are associated with each of the plurality of discrete functions of the workflow engine, and the contents of each data dictionary are specific to a selected industry, and wherein the data dictionaries associated with each task list is dynamically modifiable via the workflow designer in real time without restarting applications or rewriting underlying software programming. As recited by amended independent claim 8, the workflow

engine performs discrete functions for which all associated entry conditions evaluate to true in an order determined by the ordered task list to render a decision to a remote user.

Amended independent claim 22 recites a system for programmatically rendering a process-based decision that includes a plurality of configurable discrete tasks made available at runtime, a plurality of business parameter objects made available at runtime and capable of being embedding with any of the plurality of configurable discrete tasks for specifying automation of the process-based decisioning for a checklist, a rules database made available for configuring rule-based entry conditions and selection criteria associated with the configurable discrete tasks at runtime, an administrative interface utilized by business users at runtime for creating process categories and checklists associated with each process and for modifying the entry conditions and the selection criteria associated with the discrete tasks, a decision database for storing the process categories, the checklists, the entry conditions and the selection criteria as configured by business users at runtime, a workflow engine defined on a single platform prior to runtime for automatically processing input from a remote user and generating an instant decision based on the checklist at runtime, a dynamic data dictionary associated with each checklist formatted in XML for defining data elements and data relationships specific to a selected industry, wherein the dynamic data dictionary associated with each checklist provides a dynamic fetch and store interface with the decision database, and wherein the dynamic data dictionary for each checklist is configurable by the business users through the administrative interface at runtime to provide, translate and modify data presentation with respect to both the remote user and the workflow engine such that the workflow engine and the administrative tools can be utilized at runtime by business users across a plurality of industries at runtime without requiring restarting or reprogramming of the administrative interface or the workflow engine to customize the workflow engine and the administrative tools for relevant industries. Amended independent claim 22 further recites a messaging system for routing two-way communications between the remote user and the process administrator, the messaging system providing a digital record of programmatic transactions.

Amended independent claim 25 recites a method for workflow processing and programmatic decision-making based on object-based processes stored in memory. The method of amended independent claim 25 includes defining a plurality of configurable differentiated tasks made available at runtime, defining business parameter objects made available at runtime, defining a rules database containing logical operators for configuring rules-based entry conditions at runtime that are associated with each of the plurality of differentiated tasks, configuring a data dictionary for each of a plurality of process checklists, configuring the plurality of process checklists at runtime, receiving input from a remote source, determining programmatically an input type according to the received input, retrieving automatically a selected one of the plurality of process checklists according to the input type wherein the selected data dictionary acts as an interface between the selected process checklist and the sets of entry conditions and as an interface between the entry conditions and both the data elements and the data relationships as a function of the particular industry to which the received input corresponds, processing programmatically the received information utilizing one or more of the selected set of differentiated tasks based on the entry conditions associated with the stored process checklist, rendering an automatic decision based on the processed received information, communicating programmatically the automatic decision to the remote source and to other partners as specified by the embedded business parameter objects. The data dictionary is populated with data elements specific to a particular industry associated with a selected one of the process checklists and data relationships specific to defined software utilized for processing any of the checklists. According to amended independent claim 25, the step of configuring each process checklist includes configuring a selected set of the plurality of differentiated tasks in an ordered arrangement, configuring entry conditions associated with each of the selected set of differentiated tasks based upon logical operators from the rules database, and embedding business parameter objects with any of the selected set of differentiated tasks for configuring a degree of automation for the process checklist.

Bengston discloses a workflow distributing apparatus and method. The system of Bengston coordinates a serial, assembly-line style workflow that is executed by a plurality of

processing devices. (Bengston, col. 4, line 66 to col. 6, line 3; FIG. 1). The system of Bengston is described with reference to coordinating printing workflows, where steps of the printing process are performed on devices at disparate locations. (E.g., Bengston, col. 1, ll. 10-47; FIG. 3). The workflow is automated so as not to require any user input while in process, and the workflow continues to until completion or until an error occurs. (Bengston, Abstract; col. 11, ll. 5-8 and 27-30; col. 14, ll. 35-39; FIG. 1). Processing by the Bengston system is decentralized, and performed sequentially by a number of different processing devices that push workflows between the processing devices in a predetermined linear fashion. (Bengston, col. 11, ll. 27-31; col. 12, line 58 to col. 13, line 8; FIG. 1). Bengston distinguishes its system from other known systems that utilize centralized processors to control the flow of information. (Bengston, col. 1, line 55 to col. 2, line 4). Bengston does not disclose business parameter objects that are embedded with tasks when configuring workflow checklists for specifying automation of that particular workflow checklist. Also, as noted on page 6 of the Office Action, Bengston fails to show, teach or disclose a data dictionary or entry conditions associated with tasks or functions.

Nichols et al. discloses a tutorial system installed on and run from a local workstation for helping to teach a student new skills. (Nichols, Abstract; FIG. 1). The system of Nichols et al. provides a simulated environment that students must understand and solve themselves. (Nichols et al., Abstract). Nichols et al. discloses the use of a domain model (or data dictionary) that facilitates communication of context-specific data across generic objects of an application. (E.g., Nichols et al., col. 22, ll. 18-39). The application utilizes a fixed architecture (i.e., workflow checklist) that does not include entry conditions associated with discrete functions in the checklist. Nichols et al., however, does not show, teach or disclose the use of a data dictionaries configured for particular workflow checklists that are dynamically modifiable without restarting applications or rewriting underlying software programming. Nichols et al also does not disclose business parameter objects that are embedded with tasks when configuring workflow checklists for specifying automation of that particular workflow checklist.

Bacon et al. discloses a workflow management system that utilizes personal subflows. Bacon et al. discloses that the personal subflows can specify rule-based branch conditions (or entry conditions), and that work flow activity is determined as a function of whether particular branch conditions evaluate to "true" or "false". (Bacon et al., col. 8, ll. 16-40; col. 9, ll. 27-38; col. 10, ll. 17-45; col. 11, ln. 46 to col. 12, ln. 60). Bacon et al. specifies that the "personal subflow does not have any explicit definition of participants or agents. Instead this information is linked or bound at run-time for the personal subflow, not at definition time." (Bacon et al., col. 3, ll. 33-36; col. 9, ll. 1-6; col. 13, ll. 1-3). In other words, workflow tasks or functions (i.e., the personal subflows) are not defined prior to runtime. Instead, Bacon et al. discloses that the personal subflows are limited constructs that are unable to associate processing activity with work items, or perform activities using more than one participant. (Bacon et al., col. 9, ll. 14-19). Moreover, Bacon et al. lacks embedded business parameter objects.

The cited references fail to disclose, teach or suggest each and every limitation of amended independent claims 1, 8, 22 and 25. The amended claim language focuses more clearly on features and functionality not shown in the prior art of record, such as features and functionality of the data dictionary and business parameter objects that provide workflow-specific automation control and dynamic configuration capabilities that avoid need for programming effort to customize workflows as seen in the prior art. Thus, the rejections under §103(a) should be withdrawn. Notification to that effect is requested. Claims 2-7 and 21 depend from amended independent claim 1 and include all of the limitations of that base claim, claims 9-14 depend from amended independent claim 8 and include all of the limitations of that base claim, claims 23 and 24 depend from amended independent claim 22 and include all of the limitations of that base claim, and claims 26 and 27 depend from amended independent claim 25 and include all of the limitations of that base claim. Thus, for the reasons given above, dependent claims 2-7, 9-14, 21, 23, 24, 26 and 27 are likewise allowable over the cited art. The rejections of those dependent claims under §103(a) should be withdrawn, and notification to that effect is requested.

CONCLUSION

All of the pending claims are now in condition for allowance. The Commissioner is authorized to charge any additional fees associated with this paper or credit any overpayment to Deposit Account No. 11-0982.

Respectfully submitted,

KINNEY & LANGE, P.A.

Date: 9.14.2007 By: _____

Austen Zuege, Reg. No. 57,907
THE KINNEY & LANGE BUILDING
312 South Third Street
Minneapolis, MN 55415-1002
Telephone: (612) 339-1863
Fax: (612) 339-6580

AZ:kmm
G:\ZUEGEA\PATFILE\I194\I194.12-02\Amendment AF Aug 2007.wpd